

PIVOT clause in SQL Server

In SQL Server (Transact-SQL), the PIVOT clause allows cross tabulation to pass data from one table to another.

In SQL Server (Transact-SQL), the PIVOT clause allows cross tabulation to transfer data from one table to another, ie, get the aggregate result and move from line to column.

employee_number	last_name	first_name	salary	dept_id
12009	Sutherland	Barbara	54000	45
34974	Yates	Fred	80000	45
34987	Erickson	Neil	42000	45
45001	Parker	Sally	57500	30
75623	Gates	Steve	65000	30



TotalSalaryByDept	30	45
TotalSalary	122500	176000

The example calculates the sum and then passes the rows into columns in the data table

PIVOT clause syntax

```

SELECT cot_dautien AS ,
[giatri_chuyen1], [giatri_chuyen2], ... [giatri_chuyen_n]
FROM
() AS
PIVOT
(
    ham_tong ()
    FOR
    IN ([giatri_chuyen1], [giatri_chuyen2], ... [giatri_chuyen_n])
) AS n>; code=""> ) AS n>;

```

Variable name or variable value

cot_dautien

The column or expression will become the first column in the new table after the transition.

bidanh_cot_dautien

The name of the first column in the new table after the transfer.

giatri_chuyen1, giatri_chuyen2, . giatri_chuyen_n

List of values ??to transfer.

bang_nguon

The SELECT statement takes source data (initial data) into the new table.

bidanh_bang_nguon

The alias of bang_nguon

ham_tong

Sum functions such as SUM, COUNT, MIN, MAX or AVG.

cot_tong

The column or expression is used with ham_tong.

cot_chuyen

The column contains the value to be transferred.

bidanh_bang_chuyen

Alias ??of the table after transfer.

The PIVOT clause can be used in later versions of SQL Server: SQL Server 2014, SQL Server 2012, SQL Server 2008 R2, SQL Server 2008, and SQL Server 2005.

To follow the steps in the tutorial, see the DDL section for creating tables and DML to create data at the end of this article and then try running on your own database.

Example with the PIVOT clause

We have the table with the data as shown below.

```
so_nhan_en_production id_phong 12009 Nguyen Huong 54000 45 34974 Pham Hoa 80000 45 34987 Phan
Lan 42000 45 45001 Tran Hua 57500 30 75623 Vu Hong 65000 30
```

Run the SQL command below to create a cross-query with the PIVOT clause.

```
SELECT 'TongLuong' AS TongLuongTheoPhong,
[30], [45]
FROM
(SELECT id_phong, luong
FROM nhanvien) AS BangNguon
PIVOT
(
SUM(luong)
FOR id_phong IN ([30], [45])
) AS BangChuyen;
```

The returned result will look like the following.

TongLuongTheoPhong 30 45 TongLuong 122500 176000

The above example creates a table after the data has been transferred, indicating the total salary of the room has an ID of 30 and the room has an ID of 45. The result is on a row with 2 columns, each column is 1 room.

Specify the column in the new table of the cross-query

First need to determine which fields of information want to include in the transfer table. In this example, TongLuong is the first column, then 2 columns id_phong 30 and id_phong 45.

```
SEL ECT 'TongLuong' AS TongLuongTheoPhong ,  
[30], [ 45]
```

Determine the data in the source table

Next is the SELECT statement that will return the source data for the new table.

In this example, it is id_phong and from the table.

```
(SELECT id_phong, lu ong  
FROM nhanvien) AS Ba ngNguon
```

Need to specify the alias for the source query, in this example is BangNguon.

Determine the total calculation function

Functions that can be used in cross queries include SUM, COUNT, MIN, MAX and AVG. In this example, the sum function SUM.

```
PIVOT  
(SUM(luong)
```

Determine the value to be transferred

Finally, the value needs to be transferred to include the result. This will be the column header in the cross query.

In this example, we only need to return id_folder 30 and 45. These values ??will be the column names in the new table. Remember that these values ??are a list of limited values ??of id_phong and do not necessarily contain all values.

```
FOR id_phong IN ([30], [45])
```

DDL / DML for examples

If you have a database and want to try out the examples in the PIVOT instruction manual above, you will need DDL / DML.

DDL - Data Definition Language are table creation commands (CREATE TABLE) for use in the PIVOT clause example.

```
CREATE TA BLE phong  
( id_phong INT NOT NULL,
```

```
ten_phong VARCHAR(50) NOT NULL,  
CONSTRAINT pk_phong PRIMARY KEY (id_phong)  
) ;
```

```
CREATE TABLE nhanvien  
( so_nhanvien INT NOT NULL,  
ho VARCHAR(50) NOT NULL,  
ten VARCHAR(50) NOT NULL,  
luong INT,  
id_phong INT,  
CONSTRAINT pk_nhanvien PRIMARY KEY (so_nhanvien)  
) ;
```

DML - Data Manipulation Language are INSERT statements to create the necessary data for the table.

```
INSERT INTO phong  
(id_phong, ten_phong)  
VALUES  
(30, 'Ketoan ');
```

```
INSERT INTO phong  
(id_phong, ten_phong)  
VALUES  
(45, 'Banhang');
```

```
INSERT INTO nhanvien  
(so_nhanvien, ho, ten, luong, id_phong)  
VALUES  
(12009, 'Nguyen', 'Huong', 54000, 45);
```

```
INSERT INTO nhanvien  
(so_nhanvien, ho, ten, luong, id_phong)  
VALUES  
(34974, 'Pham', 'Flowers', 80000, 45);
```

```
INSERT INTO nhanvien  
(so_nhanvien, ho, ten, luong, id_phong)  
VALUES  
(34987, 'Phan', 'Lan', 42000, 45);
```

```
INSERT INTO nhanvien  
(so_nhanvien, ho, ten, luong, id_phong)  
VALUES  
(45001, 'Tran', 'Hue', 57500, 30);
```

```
INSERT INTO nhanvien  
(so_nhanvien, ho, ten, luong, id_phong)  
VALUES  
(75623, 'Vu', 'Hong', 65000, 30);
```

Previous article: [Query SUBQUERY child in SQL Server](#)

The following article: [Data types in SQL Server](#)

You finished reading the article "**PIVOT clause in SQL Server**" edited by the [TipsMake](#) team. We hope this article has provided you with many useful tech tips and tricks. You can search for similar articles on tips and

guides. Thank you for reading and for following us regularly.
